

# STUDI PENGGUNAAN VISUAL STUDIO 6.0 UNTUK PENGEMBANGAN SISTEM INFORMASI BERKELAS ENTERPRISE

**Djoni Haryadi Setiabudi**

Fakultas Teknologi Industri, Jurusan Teknik Informatika - Universitas Kristen Petra  
e-mail : djonihs@petra.ac.id

**Ibnu Gunawan**

Fakultas Teknologi Industri, Jurusan Teknik Informatika - Universitas Kristen Petra  
e-mail : ibnu@petra.ac.id

**ABSTRAK:** Pada penelitian ini dilakukan studi tentang sejauh mana kemampuan Visual Studio 6.0 dari Microsoft untuk pengembangan aplikasi sistem informasi yang berskala enterprise, disamping itu juga untuk mencari pedoman bagi developer khususnya Ms Visual Studio developer dalam memberikan solusi bagi aplikasi enterprise.

Metodologi yang dilakukan adalah: mempelajari MSF, mempelajari Visual Studio secara singkat, menerapkannya pada perancangan, pembuatan prototype dan pengambilan kesimpulan. Pada pengujian yang diambil sebagai contoh kasus adalah sistem informasi universitas pada sub bagian penerimaan mahasiswa baru bidang data pribadi mahasiswa.

Dari penelitian yang dilakukan dapat ditarik kesimpulan bahwa Visual Studio 6.0 enterprise edition masih terlalu kompleks untuk digunakan meskipun kekompleksan itu telah dibayar oleh kapabilitas yang tinggi dalam mengembangkan aplikasi enterprise.

**Kata kunci:** *enterprise, Visual Studio 6, Microsoft, developer, prototype.*

**ABSTRACT:** This research is about the study of capabilities and performance of Ms Visual Studio in order to develop an enterprise application beside of to search "rule of thumb" for developer especially Ms Visual Studio developer when they're plan to give solution for developing enterprise application.

The methods are: learning MSF, learning Ms Visual Studio, applying on designing and developing prototype, and making conclusions. For the testing, is used the university information system on the new student admission subsystem on student personal record.

Based on the research, can be concluded that Visual Studio 6.0a enterprise edition still too complex for used although that complexity payed by highly capability to develop enterprise application.

**Keywords:** *enterprise, Visual Studio 6, Microsoft, developer, prototype.*

## 1. PENDAHULUAN

Microsoft Visual Studio 6.0 Enterprise merupakan *tool* set yang beroperasi pada windows 2K dan NT Server untuk mengembangkan solusi berkelas *enterprise*. Diantara berbagai macam *tool* yang berkelas *enterprise*, harga dari Ms Visual Studio ini paling murah jika ditinjau dari kelengkapannya. Harganya sekitar \$1500, lebih murah dari Delphi atau C++ Builder yang harganya mencapai \$2499 dengan kelengkapan jauh dibawahnya. Selain itu, Visual Studio 6.0 ditunjang oleh *tool – tool* untuk mengembangkan aplikasi *enterprise*. Ini

merupakan nilai tambah yang tidak dipunyai oleh *software* lain sekelasnya, yang akan menghemat biaya pengembangan *software*.

Pada studi ini akan diteliti cara yang harus ditempuh, prosedur yang harus dilaksanakan, syarat yang harus dipenuhi, kemudahan penggunaan dan sampai dimana kemampuan Ms Visual Studio 6.0 Enterprise untuk pengembangan suatu aplikasi sistem informasi berbasis windows berkelas enterprise.

Yang dipakai sebagai contoh kasus adalah sistem informasi universitas secara umum. Karena besarnya ruang lingkup sistem tersebut maka hanya diambil bagian

yang cukup layak untuk dijadikan prototype yaitu di sub bagian penerimaan mahasiswa baru bidang data pribadi mahasiswa. Prototype tersebut dibuat menggunakan Visual modeller, Visual Basic dan C++ dengan bantuan MTS, SqlServer 7, DCOMCNFG dan Remote automation manager.

## 2. TEORI DASAR

### 2.1 Definisi Aplikasi Enterprise

Kata *Enterprise* bagi sebagian orang berarti “besar”, “perusahaan” dan yang sejenisnya. Sedangkan jika melihat definisi umum yang diberikan di buku *System Design and Analysis Method*, maka definisi aplikasi *enterprise* adalah koleksi dari sistem informasi yang terintegrasi secara menyeluruh yang mengakomodasi keseluruhan fungsi dasar yang dibutuhkan oleh perusahaan. Sehingga dapat diambil kesimpulan bahwa aplikasi *enterprise* adalah aplikasi komputer yang besar atau sistem aplikasi komputer perusahaan.

Jika diambil acuan dari definisi yang diberikan Microsoft pada MCSD yang berjudul *analyzing requirement and defining solution architecture*, dikatakan definisi aplikasi *enterprise* adalah aplikasi bisnis yang besar yang mempunyai kelebihan *multi user, complex, scalable, distributed* berbasis component dan mission critical. Dapat di-deploy pada *multi environment* seperti intranet, internet, network lintas platform, bersifat *data centric* dan dapat memenuhi syarat security yang lengkap. Secara singkat dikatakan bahwa aplikasi *enterprise* merupakan sistem yang sangat kompleks. Fitur yang termasuk di dalam aplikasi *enterprise* adalah: kompleks, berorientasi bisnis, *mission critical*.

### 2.2 MS Visual Studio 6.0 Enterprise

Visual Studio adalah kumpulan development tools dari Microsoft untuk membangun aplikasi *enterprise* dan kelengkapannya. Visual Studio mempunyai 5 tools primer yaitu Visual Basic, Visual C++, Visual Interdev, Visual Foxpro, dan Visual J++. Visual Studio tersedia dalam 2 edisi yaitu edisi professional dan *enterprise*.

Edisi *enterprise* mempunyai berbagai macam *tools* lain yang tidak dipunyai edisi professional yaitu SQL server developer edition, MTS (Microsoft Transaction Server), Visual Source Safe, Visual Component Manager, Visual Modeler, Visual Studio Analyzer, T-SQL Remote Debugger, Visual APE (*Application Performance Explorer*), Visual Database Tools, SNA Server, dan dukungan untuk Resident RISC processor serta MSDN (Microsoft Developer Network).

### 2.3 Tahapan Untuk Pengembangan Aplikasi Enterprise

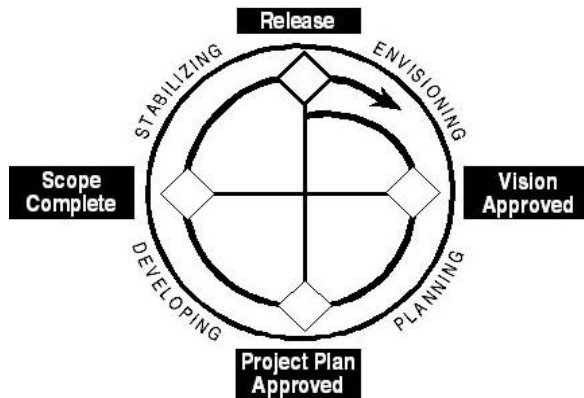
Tahapan ini didasarkan pada MSF (*Microsoft Solution Framework*) yang terdapat pada buku *Analyzing Requirement and Defining Solution Architecture: MCSD (Microsoft Certified Solution Developer) exam 070 –100*.

Dipilih MSF karena produk pembuat *software* aplikasi yang digunakan adalah produksi Microsoft, sehingga akan sangat cocok jika metoda yang digunakan adalah metoda buatan Microsoft. Beberapa MSF yang perlu diperhatikan dalam pengembangan aplikasi *enterprise* adalah :

- MSF Enterprise Architecture model.  
Model ini menyediakan urutan yang konsisten yang akan membantu untuk mengembangkan aplikasi *enterprise* secara cepat (RAD) sampai versioned releases. Model ini mensejajarkan teknologi informasi dengan *business requirements* melalui 4 *perspectives: business, Application, Information, and Technology*. Dengan menggunakan model ini maka *planning cycle* dari *architecture enterprise* dapat dipersingkat.
- MSF Process Model.  
Dalam mengembangkan aplikasi *enterprise* ada proses yang harus dilalui, Microsoft mengistilahkannya MSF *Process* model. Yang selanjutnya dapat dilihat pada Gambar 1.

MSF Development Process Model terdiri dari 3 bagian yaitu: 4 *phased process* (*Envisioning, planning, developing, stabilizing*), 4 *milestone-driven process* (*vision*

*approved, project plan approved, scope complete, released*). An iterative process yaitu berulangnya ke 4 *phased* dan *milestone-driven process* tersebut.



**Gambar 1. MSF Process Model**

## 2.4 Microsoft DNA (Distributed I-Net Architecture)

Dikenal juga dengan nama *MSF Application model*, atau *3 tiered architecture*. Berguna sebagai pedoman utama dalam mengembangkan *logical* dan *physical design*. Yang dimaksud dengan I-Net adalah InterNet dan IntraNet. Model ini dibagi menjadi 3 submodel, berikut ini adalah ke-3 submodel tersebut beserta contoh implementasinya di kehidupan sehari-hari :

- *User Service model*, contoh: webbrowser, aplikasi win 32.
- *Business Service model*, contoh: MTS
- *Data Service model*, contoh: SQL Server.

Untuk menunjang keberhasilan membentuk DNA ini maka diperlukan permodelan sistem sehingga akan mempermudah *developer* untuk mendevlop fungsi, *object*, maupun *service* yang diperlukan.

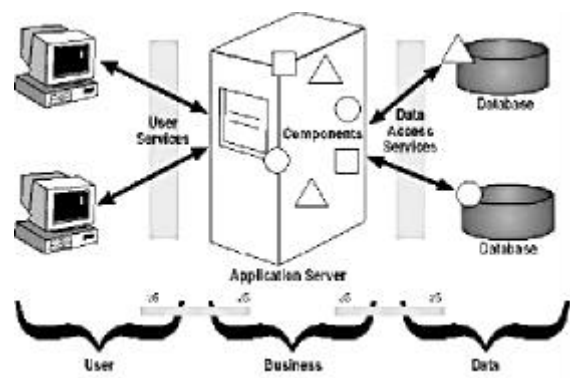
Pada arsitektur *three tier*, fungsi *Trigger*, *Error handling* dan *Login database* serta pemanggilan *stored procedure* diterapkan pada *tier* bisnis sedangkan *tier server* hanya berisi tabel, *stored procedure* dan konstrain. Untuk *tier user* berisi *User interface* untuk login dan hubungan dengan tier bisnis.

Sehingga dapat ditarik kesimpulan bahwa keuntungan *Three tier* dibandingkan *two tier* adalah aplikasi sistem *Three tier* lebih scalable dan flexible. Seandainya database konsultan tersebut harus melayani sistem yang mempunyai *Multiple database server*

maka *developer* harus merombak keseluruhan aplikasi pada client dan juga server. Sedangkan untuk *Three tier* hanya perlu perubahan pada tier bisnis/*middle tier*. Keuntungan yang lain yaitu *Three tier* lebih aman dibandingkan *two tier* sebab rule bisnis dan *Login database* diletakkan pada *middle tier* dimana hanya pemrogram saja yang mengetahui bagaimana cara mengedit dibandingkan dengan *two tier* dimana bisnis rule bisnis dan *Login database* diletakkan pada database dan client dimana DB admin dan user yang berpengalaman dapat mengacaukannya.

## 3. PERANCANGAN PROTOTYPE APLIKASI

Sesuai dengan MSF, arsitektur yang dipilih untuk pembuatan sistem adalah 3 tier atau Windows DNA seperti pada Gambar 2. Sesuai dengan Gambar 2, diperlukan 3 komputer untuk implementasi sistemnya yaitu untuk data server, application server dan komputer user. Pada data server, digunakan software SQL server 7 yang dioperasikan pada NT server. Untuk application servernya dalam hal ini adalah MTS maka diperlukan Win NT sebagai OS dasarnya dengan minimum service pack 3. Catatan yang perlu diperhatikan bahwa MTS tidak akan berjalan secara lengkap pada Win9.x



**Gambar 2. Arsitektur 3 tier / Windows DNA**

Komputer user/client kebutuhan minimumnya adalah bisa menjalankan Windows 98 untuk aplikasi yang bersifat win32 dan minimum bisa menjalankan browser internet untuk aplikasi yang bersifat I-Net. Untuk pengembangan aplikasinya sendiri sebenar-

nya bisa dengan menggunakan 1 komputer dengan catatan komputer tersebut cukup mampu untuk bertindak sebagai data server sekaligus application server. Minimum requirementsnya yang perlu diperhatikan adalah memiliki prosesor PII-350 dengan cache 512 Kb, memori 128Mb beroperasi pada 100Mhz FSB, Harddisk 4.3Gb, sehingga bisa menjalankan minimum Win NT tanpa tersendat sendat.

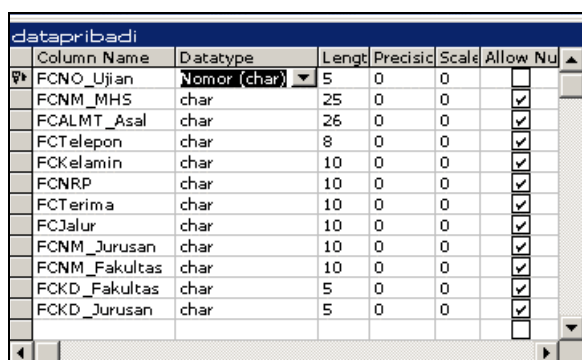
Dalam Visual Studio terdapat 3 macam bahasa pemrograman yaitu Java, Basic, dan C++. Ke-3 nya saling terintegrasi satu dengan yang lain. Karena Visual Basic sangat mendukung RAD dan dikembangkan pertama kali sebagai bahasa pemrograman yang dikonsentrasikan untuk database, maka dipilihlah Visual basic sebagai bahasa utama untuk pengembangan aplikasi enterprise ini dengan catatan bahwa aplikasi yang berjalan pada komputer klien adalah aplikasi Win 32 Native dan untuk load balancer pada mid tier menggunakan Visual C++.

Aplikasi enterprise yang akan dibuat secara keseluruhan terdiri dari 3 buah yang terdiri dari 1 buah aplikasi untuk masing - masing layer. Yaitu database pada datalayer, 2 komponen yang saling berinteraksi pada mid tier dan Win 32 clients application pada user layer.

### 3.1 Pembuatan Data Layer

Database contoh yang akan dibuat terdiri dari 3 tabel yang saling terhubung satu dengan yang lain dan 1 stored procedure. Stored procedure tersebut bernama "semua\_mahasiswa" dengan sintaks sebagai berikut:

```
create procedure semua_mahasiswa
as SELECT * FROM datapribadi
```

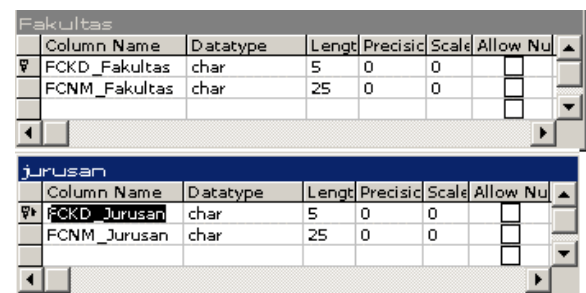


Column Name	Datatype	Length	Precision	Scale	Allow Null	Index
FCNO_Ujian	Nomor (char)	5	0	0	0	No
FCNM_MHS	char	25	0	0	0	Yes
FCALMT_Asal	char	26	0	0	0	Yes
FCTelepon	char	8	0	0	0	Yes
FCKelamin	char	10	0	0	0	Yes
FCNRP	char	10	0	0	0	Yes
FCTerima	char	10	0	0	0	Yes
FCJalur	char	10	0	0	0	Yes
FCNM_Jurusan	char	10	0	0	0	Yes
FCNM_Fakultas	char	10	0	0	0	Yes
FCKD_Fakultas	char	5	0	0	0	Yes
FCKD_Jurusan	char	5	0	0	0	Yes

**Gambar 3. Isi Tabel Data Pribadi**

Sedangkan tabel yang dibuat masing masing bernama data pribadi, fakultas dan jurusan. Data pribadi berisi 12 field yang seluruhnya mempunyai data type char. Untuk lebih jelasnya dapat dilihat pada Gambar 3.

Tabel selanjutnya adalah fakultas dan jurusan. Fakultas berisi 2 field yang seluruhnya mempunyai data type char. Sedangkan jurusan berisi 2 field yang seluruhnya mempunyai data type char. Untuk lebih jelasnya dapat dilihat pada Gambar 4.



Column Name	Datatype	Length	Precision	Scale	Allow Null	Index
FCKD_Fakultas	char	5	0	0	0	No
FCNM_Fakultas	char	25	0	0	0	No

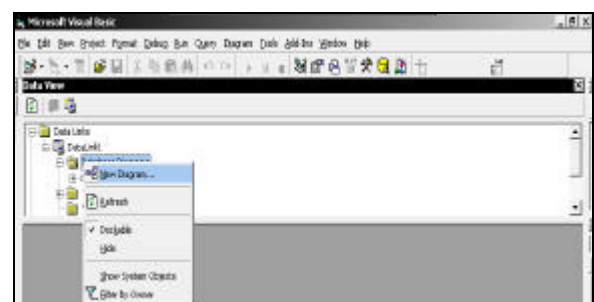
  

Column Name	Datatype	Length	Precision	Scale	Allow Null	Index
FCKD_Jurusan	char	5	0	0	0	No
FCNM_Jurusan	char	25	0	0	0	No

**Gambar 4. Isi Tabel Fakultas dan Jurusan**

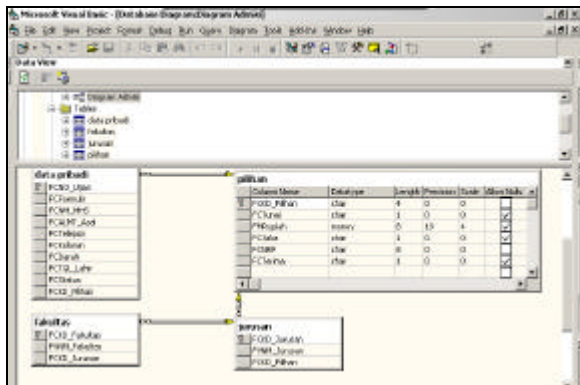
Data layer sebenarnya adalah database relasional yang ada di MSSQL Server (solusi Microsoft untuk RDBMS). Dapat dikembangkan dengan melalui SQL Server Enterprise Manager, atau Visual Database tools – Ms Visual Studio.

Pembuatan Database dapat dimulai dengan membuat tabel yang diperlukan. Caranya adalah dengan mengklik kanan mouse pada tab tables setelah datalink diexpand. Dalam hal ini ada 4 buah tabel yang akan dibuat yaitu: Tabel data pribadi berisi data pribadi calon mahasiswa, Tabel pilihan berisi data pilihan jurusan serta kelengkapannya yang dipilih oleh calon mahasiswa, Tabel jurusan berisi data jurusan yang ada di universitas yang bersangkutan dan Tabel fakultas berisi data fakultas yang ada di universitas yang bersangkutan.



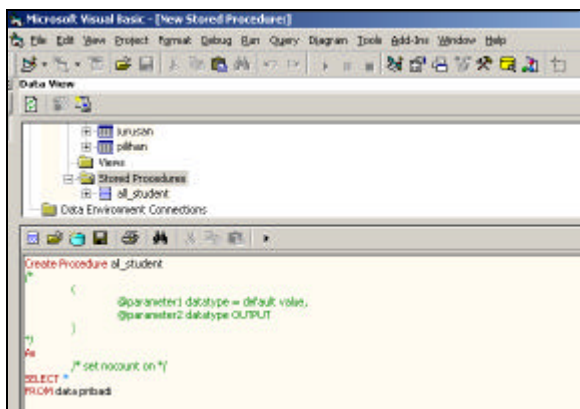
**Gambar 5. Pembuatan Diagram**

Setelah tabel dibuat, diset relasi antar database dengan menggunakan database diagram, dengan cara meng-klik kanan pada database diagram dan pilih new seperti terlihat pada Gambar 5. Diagram yang selesai dibuat dapat dilihat pada Gambar 6.



### Gambar 6. Database Diagram

Pembahasan dilanjutkan pada stored procedure. Stored procedure berguna untuk meningkatkan bandwidth dari middle layer ke data layer. Hal ini akan meningkatkan efisiensi. Stored Procedure yang akan dibuat adalah stored procedure yang sangat sederhana yaitu untuk mengambil semua data dari data pribadi mahasiswa. Seperti yang terlihat pada Gambar 7, diawali dengan inisialisasi RPC (Remote Procedure Call) dengan menggunakan automation manager dan diakhiri dengan windows untuk debug.



### Gambar 7. Pembuatan Stored Procedure

### 3.2 Pembuatan Business Layer (Mid Tier)

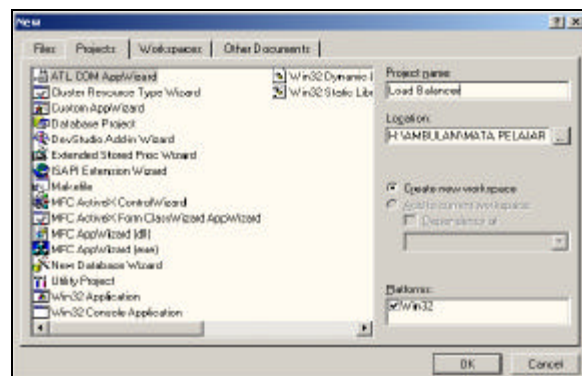
Kelanjutan dari data layer adalah middle layer atau yang lebih dikenal dengan business layer. Inti dari layer ini adalah MTS dimana bertugas untuk: Login pada database, Menjalankan query, menentukan

business rule dan Error Handling dari database. Ada 2 buah komponen yang perlu dibuat, yaitu Load Balancer dan Server Mahasiswa.

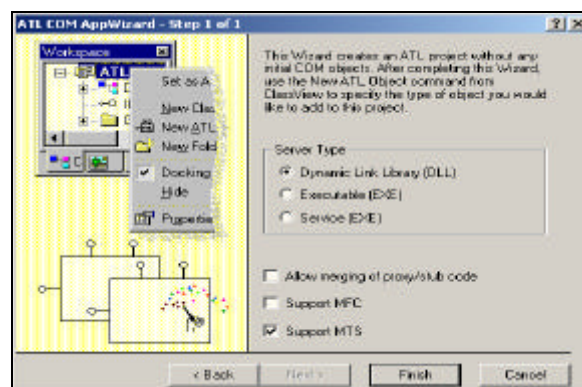
- Load Balancer

Sebagai catatan, komponen ini merupakan komponen yang telah diprogram ulang sedemikian rupa sehingga bisa berjalan dengan baik pada sistem ini. Komponen asli merupakan komponen load balancer dari buku Building Enterprise Solution With Visual Studio 6.0. Dibuat dengan menggunakan Visual C++ memakai fitur ATL.

Cara pembuatannya adalah pada Visual C++ 6.0 Enterprise. Dipilih new project, dan highlight ATL COM AppWizard. Diketikkan nama Load Balancer pada project name seperti yang terlihat pada Gambar 8. Untuk wizard window selanjutnya pilihlah option untuk support MTS seperti pada Gambar 9. Setelah itu diklik finish agar wizard mengautogenerate kode program secara mendasar.



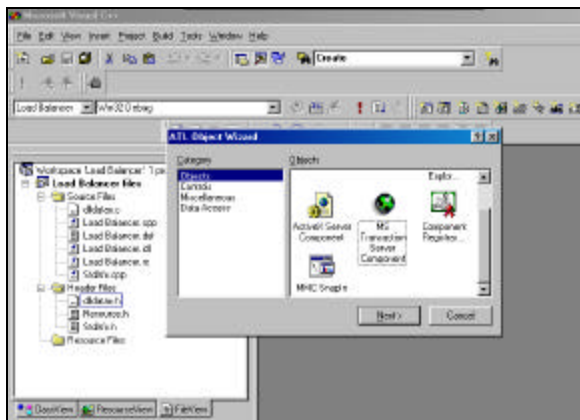
### Gambar 8. Pembuatan Komponen MTS Dengan Visual C++



**Gambar 9. Option yang Dipilih Untuk Komponen MTS di Visual C++**



Dilanjutkan dengan membuat object bagi MTS tersebut. Cara untuk membuat objek tersebut adalah dengan memilih menu insert New ATL Object. Dipilih MS Transaction server komponen seperti ada di Gambar 10.

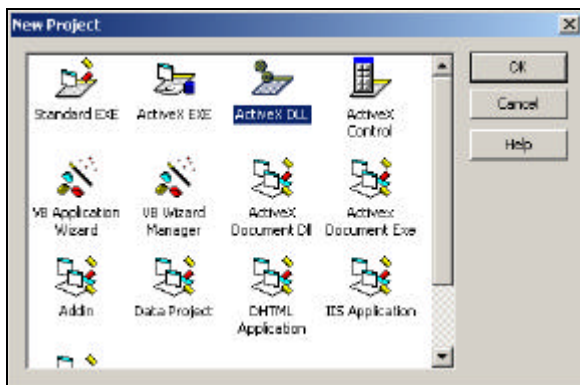


**Gambar 10. Pembuatan Object untuk MTS Menggunakan ATL**

- Server Mahasiswa

Komponen ini merupakan komponen yang telah diprogram ulang sehingga bisa berjalan dengan baik pada sistem ini. Komponen ini diharapkan dan diasumsikan untuk menjadi “server” data pada mid tier yaitu MTS. Bertugas untuk menjadi jembatan antara client dan server database pada pusat data. Juga sebagai pelaksana untuk beberapa fungsi dasar database seperti insert, delete, dan, refresh data dari server.

Karena itulah maka dibuat menggunakan Visual Basic. Dan cara pembuatannya dimulai dengan memilih new project pada menubar Visual Basic. Dan mengambil pilihan ActiveX DLL seperti terlihat pada Gambar 11.

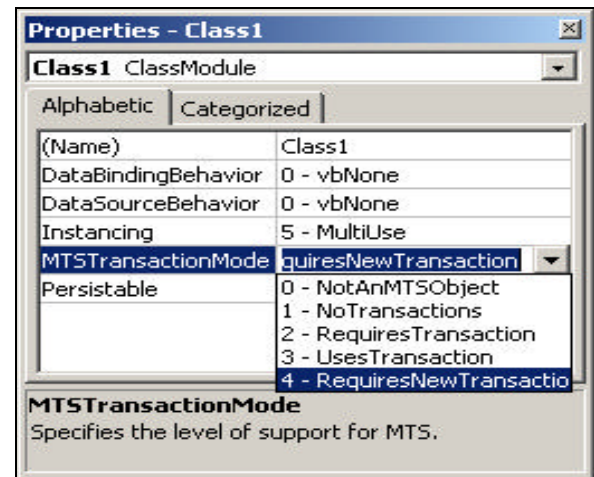


**Gambar 11. Memulai Membuat Komponen MTS pada VB**

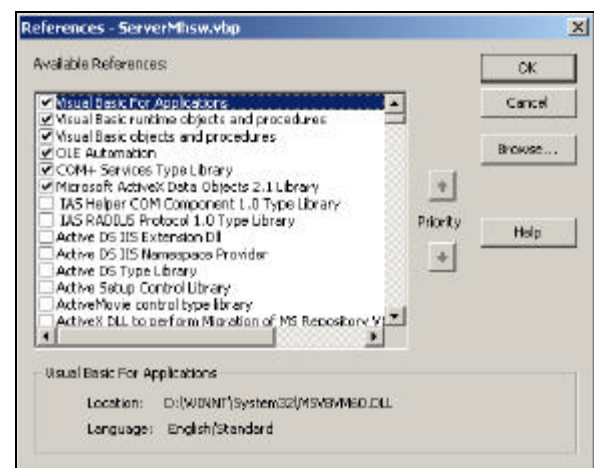
Lalu dilanjutkan dengan menset properties MTS pada object properties ke Require

New Transaction seperti pada Gambar 12. Kemudian ditambahkan referensi untuk COM+ type library dan Microsoft Active Data Object 2.x library, hal ini dapat dilihat pada Gambar 13.

Setelah semua persiapan selesai dilakukan maka coding dapat dimulai dengan menambahkan option explicit pada bagian deklarasi dan disertai dengan pendeklarasian object control dan context sebagai interface utama bagi MTS yang kemudian diikuti dengan pendeklarasian ADODB.Connection sebagai awal untuk memulai fungsi database pada komponen.



**Gambar 12. Setting Properti Kelas Untuk MTS**

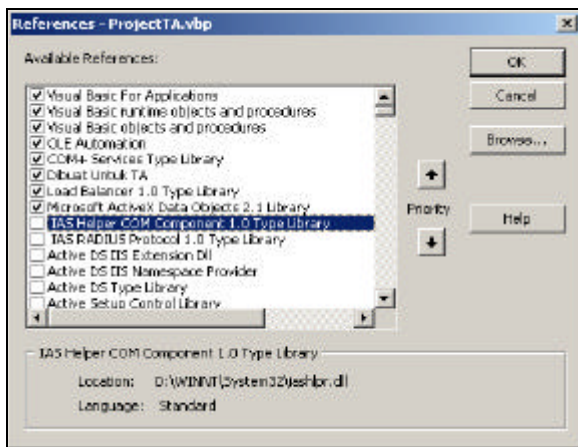


**Gambar 13. Penambahan Referensi untuk Kelas MTS Pada VB**

### 3.3 Pembuatan User Layer

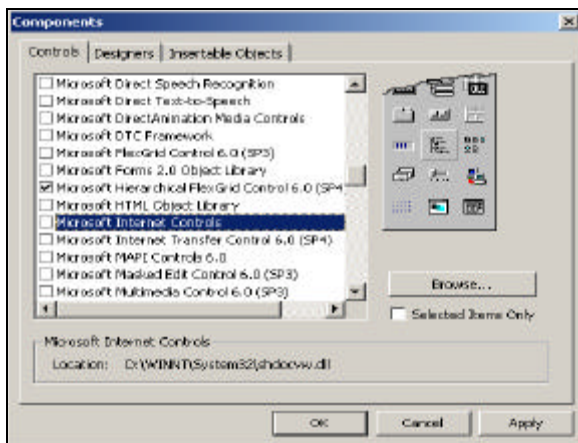
Pada dasarnya pembuatan aplikasi ini seperti aplikasi biasa hanya saja ada beberapa tambahan seperti referensi dan pendeklarasian object baru. Pertama dipilih

new project pada menubar lalu dipilih Standard EXE dan diikuti dengan menentukan referensi pada 2 komponen MTS yang baru dibuat seperti pada Gambar 14.



**Gambar 14. Setting Referensi Untuk User tier**

Ada beberapa komponen tidak dasar (tidak terdapat pada toolbox pada waktu pertama kali VB dijalankan) yang dipakai, untuk itu harus ditambahkan dengan menekan ctrl dan dilanjutkan dengan tombol t. Ditambahkan juga komponen MSH Flex Grid seperti pada Gambar 15.



**Gambar 15. Penambahan Referensi Untuk MSH FlexGrid Control**

Selanjutnya dapat memulai melakukan coding pada user layer ini, seperti biasa dilakukan inisialisasi sebagai berikut :

```
Dim objMahasiswa As clsMahasiswa
```

```
Dim rsGeneric As New ADODB.Recordset  
' Generic Recordset
```

```
Dim rsMahasiswa As ADODB.Recordset  
' Mahasiswa Recordset
```

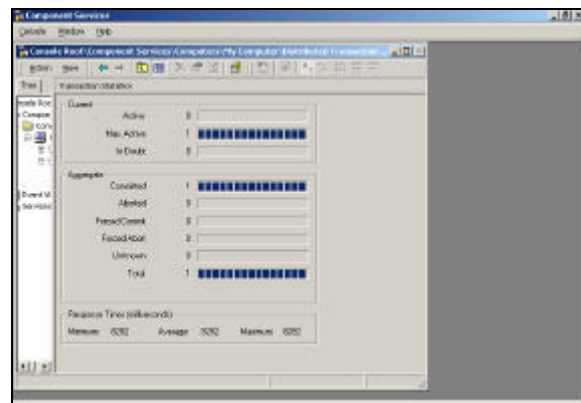
```
Dim ID_Mahasiswa As String
```

Lalu untuk hal-hal yang berhubungan dengan komponen-komponen yang terletak pada form misalnya pada waktu dilakukan penekanan pada tombol ambil data mahasiswa terbaru yang terjadi adalah VB akan membawa programmer untuk melakukan coding sesuai dengan even yang terjadi.

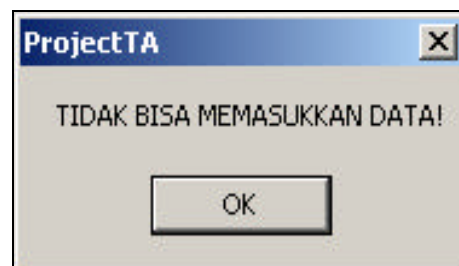
#### 4. PENGUJIAN

Untuk menguji apakah benar proyek ini memakai sistem 3 layer, dapat dibuka MTS explorer atau component service pada bagian transaction statistic, yaitu pada mid tier server. Jika benar maka akan tercatat beberapa transaksi yang dipergunakan. Hal ini dapat diperlihatkan melalui Gambar 16. dimana paling tidak ada 1 transaksi yang terjadi.

Transaksi ini akan terus bertambah jika user mempergunakan program, misalnya ketika user menambah data baru maka program akan mengecek apakah data tersebut sudah ada pada SQL server atau tidak berdasarkan FCNO ujian. Jika FCNO ujian sama maka transaksi akan dibatalkan dan pesan error akan keluar. Hal ini dapat dilihat pada Gambar 17 dan Gambar 18.

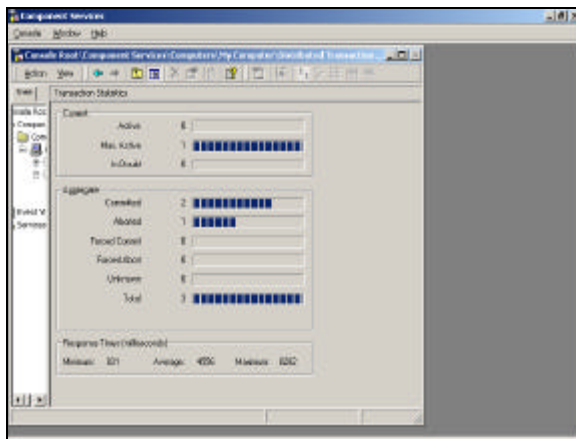


**Gambar 16. Pencatatan Transaksi Yang Terjadi**



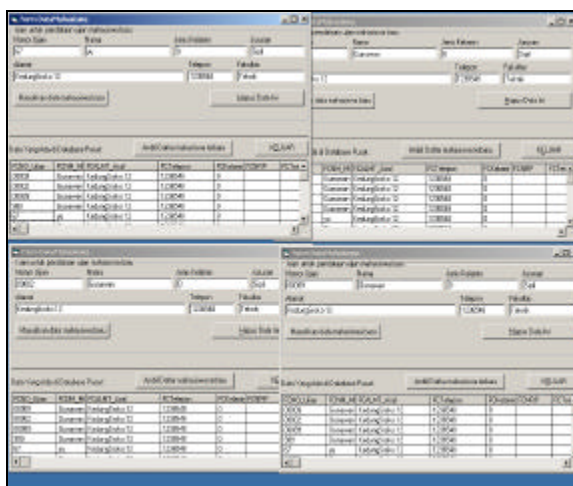
**Gambar 17. Pesan Error yang Ditampilkan**

Selanjutnya uji coba dilanjutkan dengan memperhatikan transaksi yang terjadi dan pesan error yang ditampilkan. Dengan catatan bahwa setiap transaksi yang terjadi pasti akan dicatat pada transaction list baik itu yang berhasil maupun gagal. Jika tidak ada campur tangan admin maka tidak akan ada transaksi yang dipaksa commit atau aborted.



**Gambar 18. Transaksi Yang Dibatalkan**

Perlu juga dilakukan pengujian yang melibatkan banyak client dimana merupakan salah satu fitur daripada aplikasi enterprise. Pengujian ini dilakukan dengan menggunakan 4 client program yang berjalan secara simultan pada sebuah komputer. Transaksi yang terjadi tidak bersamaan sehingga mencegah terjadinya mekanisme atomicity. Gambar dari layar pengujian ini dapat dilihat pada Gambar 19.



**Gambar 19. Pengujian 4 Transaksi Secara Simultan**

## 5. KESIMPULAN

- Untuk menilai suatu aplikasi memiliki level enterprise atau tidak semata-mata bisa didasarkan pada definisi Microsoft tentang aplikasi enterprise.
- Program prototype yang dibuat belum keseluruhan memenuhi definisi dasar dari suatu aplikasi enterprise.
- Bahwa MSF merupakan suatu pedoman pokok yang dapat dipakai dan disarankan untuk dipakai jika memakai program Microsoft untuk membuat solusi berkelas enterprise.
- Visual Studio hanya benar benar bisa mengeluarkan seluruh kemampuannya jika diinstall pada NT Server atau Win2K.

## DAFTAR PUSTAKA

1. Benage, Don et al., *Building Enterprise Solution With Visual Studio 6.0.*, Indianapolis: Sams Publishing, 1999.
2. Byrne, Jefry L., *Microsoft SQL Server What Database Administrator Need to Know*, NewYork : Prentice Hall, 1997.
3. Chapman, Davis, *Teach Yourself Visual C++ 6 in 21 Days*, Indianapolis: Sams Publishing, 1998.
4. Gunawan, Ibnu., *Kajian Penggunaan Ms Visual Studio 6.0a Enterprise Edition untuk Pengembangan Aplikasi Enterprise Information System*, Surabaya: Universitas Kristen Petra, 2001.
5. Pery, Greg, *Teach Yourself Visual Basic 6 in 21 Days*, Indianapolis: Sams Publishing, 1998.
6. Sullivan, G.A., *Special Edition Using Visual Studio 6.0*, Indianapolis: Que Corporation, 1997
7. Williams, Charles, *Proffesional Visual Basic 6 Databases with MTS and ASP*, Chicago: Wrox Ltd, 1998
8. Whitten, Jeffrey L., *System Analysis and Design Methods*, California: McGraw Hill, 2001



9. \_\_\_\_\_: *Analyzing Requirement and Defining Solution Architecture*, MCSD (Microsoft Certified Solution Developer) training kit, Washington: Microsoft Press 1999.